

Applicant : David Corboy  
Serial No. : 08/866,857  
Filed : May 30, 1997  
Page : 3

Attorney's Docket No.: 00975-124001 / Multimedia 15

### REMARKS

In view of the foregoing amendments and following remarks, reconsideration and allowance of this application are requested. Claims 1-16 and 31-52 are pending, with claims 1 and 10 being independent. Claims 51 and 52 have been added.

### Interview Summary

Applicant would like to thank Examiner Huynh and Primary Examiner Hong for the courtesies extended to Applicant's representative during the personal interview conducted on October 17, 2001. The foregoing amendments and the following remarks reflect the substance of the interview.

### 35 U.S.C. § 103(a) Kauffman/Berry et al./Boezeman et al. Rejection

Claims 1, 4, 9-14, and 31-50 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,586,235 ("Kauffman") in view of U.S. Patent No. 5,692,205 ("Berry") and U.S. Patent No. 5,889,519 ("Boezeman"). Claims 4, 9, and 31-40 depend from claim 1. Claims 11-14 and 41-50 depend from claim 10. This rejection, insofar as it pertains to the presently pending claims, is respectfully traversed.

Independent claims 1 and 10 are directed to producing a streaming multimedia document and recite "choreographing information comprising data slices from the first file support object and the second file support object multiplexed so as to incrementally render the first file support object and the second file support object to the user at substantially the same time." Neither Kauffman, Berry, nor Boezeman, alone or in combination, teach or suggest at least this feature of independent claims 1 and 10.

Kauffman is directed to an interactive multimedia system and method that provides a standard document structure for organizing and storing information. See Kauffman at col. 2, ll. 35-39. The document includes at least one page, and the page contains at least one asset file and a page execution script. See Kauffman at col. 2, ll. 39-51; col. 6, l. 50. Authoring code is used to create the page execution scripts, which instruct when to display or play information contained

in the asset file. See Kauffman at col. 12, ll. 62-65. However, Kauffman fails to describe or suggest choreographing information comprising data slices from the first file support object and the second file support object multiplexed so as to incrementally render the first file support object and the second file support object to the user at substantially the same time, as recited in claims 1 and 10.

Berry is directed to a method and system for integration of multimedia presentations within an object oriented user interface, which provides for encapsulating multimedia data within an object. See Berry at col. 2, ll. 41-45. Berry does not require separate audio or video objects to be manipulated by the user. See Berry at col. 2, ll. 41-45. However, Berry does not remedy the failure of Kauffman to describe or suggest choreographing information comprising data slices from the first file support object and the second file support object multiplexed so as to incrementally render the first file support object and the second file support object to the user at substantially the same time, as recited in claims 1 and 10.

Boezeman is directed to a method and system for a multimedia application development sequence editor using a wrap corral, which allows parts of a multimedia title to be continuously played or shown. See Boezeman at col. 2, ll. 41-46. Boezeman describes a sequence editor user interface tool for synchronizing objects. See Boezeman at col. 5, ll. 40-50. However, Boezeman does not remedy the failure of Kauffman and Berry to describe or suggest choreographing information comprising data slices from the first file support object and the second file support object multiplexed so as to incrementally render the first file support object and the second file support object to the user at substantially the same time, as recited in claims 1 and 10.

The Examiner's rejection asserts that Boezeman discloses "the choreographing information as defined in the timing comprising data slices from the first file support object interleaved with data slices from the second file support object so as to incrementally display the first file support object and the second file support object to the user." As support, the Examiner's rejection cites three portions of Boezeman: (1) col. 1, lines 50-60; (2) col. 5, lines 4-19; and (3) col. 5, lines 30-55. The cited portions of Boezeman, however, fail to support the assertion.

First, col. 1, lines 50-60 of Boezeman states:

One of the most difficult problems that face creators/developers of multimedia titles is synchronization of time and events. For example, a developer must be able to synchronize audio with a video presentation and provide images that appear and disappear over time. In many complex real world examples it is desirable for two or more multimedia items to be synchronized based on one or more of the following: an event; a relative time relationship (i.e., A before B, A after B, A coincident with B); and an absolute time. One product attempting to deal with time synchronization

While recognizing the problem of synchronization, this citation to Boezeman does not describe or suggest choreographing information comprising data slices from the first file support object and the second file support object multiplexed so as to incrementally render the first file support object and the second file support object to the user at substantially the same time, as recited in claims 1 and 10.

Second, col. 5, lines 4-19 of Boezeman states:

animation to play; 2) during the course of the animation, a piece of audio is also played; 3) at the simultaneous ending of the animation and audio, a video plays, the second half of which plays at twice the rate of the first half; 4) independent of 1-3 above, an image appears ten seconds after the push button is pressed, remains visible for 20 seconds while moving across the screen, and then disappears; and 5) shortly after the video begins an exit push button appears on the display.

Referring to FIG. 16, a real world example that might require this type of processing could be a computer based training application (a multimedia title 300) for a student in an automotive service center. The student would click on an arbitrary part, a push button 302, on a computer to view information about how to assemble a previously selected automobile part. The arbitrary event, i.e., clicking the push

This portion of Boezeman does not describe or suggest choreographing information comprising data slices from the first file support object and the second file support object multiplexed so as to incrementally render the first file support object and the second file support object to the user at substantially the same time, as recited in claims 1 and 10.

Third, col. 5, ll. 30-55 of Boezeman states:

completeness of the student's background. Due to time constraints, the second half of the video plays at twice the rate of the first half. Independent of the above processing, an image (image part 312) is displayed 10 seconds into the playing of the animation, and moves across the screen for 20 seconds. The image is a welcome message wishing the student good luck. Also independent of the above processing, shortly after the video begins playing, an exit push button (push button part 314) appears at the bottom of the screen allowing the student to exit the video if it is deemed unnecessary to further understanding.

To solve the synchronization problem as described above, a sequence editor in accordance with the present invention is provided. The sequence editor is a mechanism that allows a user to synchronize relative time, absolute time and event time together in an easy to use manner. The solution assumes that items for synchronization (such as an animation, video, audio, image) have been previously identified and selected by a user. There are several tools which allow selection of these types of items for synchronization and further description is not required.

The sequence editor approach to synchronization starts with events. Events are unpredictable and often occur randomly. They can be caused by user interaction or other asynchronous activity. Examples of events could be a push button being clicked, a video playing to completion, an

This portion of Boezeman does not describe or suggest choreographing information comprising data slices from the first file support object and the second file support object multiplexed so as to incrementally render the first file support object and the second file support object to the user at substantially the same time, as recited in claims 1 and 10.

Independent claims 1 and 10 thus are allowable for at least this reason. Claims 4, 9, 11-14, and 31-50 are allowable by virtue of their dependency, as well as on their own merits.

Accordingly, reconsideration and withdrawal of this rejection are respectfully requested.

**35 U.S.C. § 103(a) Kauffman/Berry/Boezeman Rejection**

Claims 31-50 also stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Kauffman, Berry, and further in view of Boezeman. Claims 31-40 depend from claim 1 and claims 41-50 depend from claim 10. This rejection, insofar as it pertains to the independent claims, is respectfully traversed.

Boezeman fails to remedy the deficiencies of Kauffman and Berry with respect to independent claims 1 and 10. Claims 31-50 thus are allowable by virtue of their dependency, as well as on their own merits.

Accordingly, reconsideration and withdrawal of this rejection are respectfully requested.

**35 U.S.C. § 103(a) Kauffman/Berry/Boezeman/Ando Rejection**

Claims 2, 3, 7, and 8 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Kauffman, Berry, and Boezeman, and further in view of U.S. Patent No. 5,600,826 to Ando (“Ando”). Claims 2, 3, 7, and 8 depend from claim 1. This rejection, insofar as it pertains to the independent claims, is respectfully traversed.

Ando is directed to a structured data processor for converting between sequential and tree structured data, including a structured data treating unit for editing data. See Ando at col. 4, ll. 25-43; col. 6, ll. 44-47. Ando fails to remedy the deficiencies of Kauffman, Berry, and Boezeman with respect to independent claim 1. Claims 2, 3, 7, and 8 thus are allowable by virtue of their dependency, as well as on their own merits.

Accordingly, reconsideration and withdrawal of this rejection are respectfully requested.

**35 U.S.C. § 103(a) Kauffman/Berry/Boezeman/Johnson Rejection**

Claims 5 and 6 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Kauffman, Berry, and Boezeman, and further in view of U.S. Patent No. 5,892,847 to Johnson (“Johnson”). Claims 5 and 6 depend from claim 1. This rejection, insofar as it pertains to the independent claims, is respectfully traversed.

Johnson is directed to a method and apparatus for compressing images, including an encoder that creates a file format that layers the compressed image. See Johnson at col. 4, ll. 30-49. Johnson fails to remedy the deficiencies of Kauffman, Berry, and Boezeman with respect

Applicant, : David Corboy  
Serial No. : 08/866,857  
Filed : May 30, 1997  
Page : 8

Attorney's Docket No.: 06975-124001 / Multimedia 15

to independent claim 1. Claims 5 and 6 thus are allowable by virtue of their dependency, as well as on their own merits.

Accordingly, reconsideration and withdrawal of this rejection are respectfully requested.

**35 U.S.C. § 103(a) Kauffman/Berry/Boezeman/Brown Rejection**

Claims 15 and 16 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Kauffman, Berry, and Boezeman, and further in view of Brown (Using Netscape 2, 1995) ("Brown"). Claims 15 and 16 depend from claim 10. This rejection, insofar as it pertains to the independent claims, is respectfully traversed.

Brown is directed to the use of frames in an HTML document. See Brown at 773-777. Brown fails to remedy the deficiencies of Kauffman, Berry, and Boezeman with respect to independent claim 10. Claims 15 and 16 thus are allowable by virtue of their dependency, as well as on their own merits.

Accordingly, reconsideration and withdrawal of this rejection are respectfully requested.

**Additional Claims 51 and 52**

Claims 51 depends from claim 1 and 52 depends from claim 10. Thus, claims 51 and 52 are allowable by virtue of their dependency, as well as on their own merits.

Applicant : David Corboy  
Serial No. : 08/866,857  
Filed : May 30, 1997  
Page : 9

Attorney's Docket No.: 06975-124001 / Multimedia 15

**CONCLUSION**

Attached is a marked-up version of the changes being made by the current amendment.

Applicant submits that all of the claims are in condition for allowance. A check in payment of excess claims fees required by the above amendment is enclosed. Please apply any other charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: November 9, 2001

  
\_\_\_\_\_  
Scott R. Boalick  
Reg. No. 42,337

Fish & Richardson P.C.  
601 Thirteenth Street, NW  
Washington, DC 20005  
Telephone: (202) 783-5070  
Facsimile: (202) 783-2331

**Version with markings to show changes made**

**In the claims:**

Claims 1 and 10 have been amended as follows:

1. A method for producing a hierarchical data file for a streaming multimedia document having a plurality of object files, the data file having different file formats encapsulated within the data file, the method comprising:

encapsulating in a multimedia document a first file support object including information in a first file format;

supporting the first file support object by the multimedia document;

encapsulating in the multimedia document a second file support object including information in a second file format;

supporting the second file support object by the multimedia document;

encapsulating in the multimedia document choreographing information for allowing a document author to define the timing at which the first file support object and the second file support object are retrieved by a user, the choreographing information comprising data slices from the first file support object [interleaved with data slices from] and the second file support object multiplexed so as to incrementally [display] render the first file support object and the second file support object to the user at substantially the same time.

10. A computer system having a memory storing a hierarchical data file structure that encapsulates a plurality of different file formats to form a streaming multimedia document having a plurality of object files, the multimedia document being capable of being displayed on a display of a computer system, the data file comprising:

a document including information for controlling the display;

a first support object including information in a first file format, the first support object being encapsulated in the document and being capable of supporting a plurality of first

lower objects, each first lower object being a lower level object than the first support object in the hierarchical data file structure;

a second support object including information in a second file format, the second support object being encapsulated in the document and being capable of supporting a plurality of second lower objects, each second lower object being a lower level object than the second support object in the hierarchical data file structure; and

choreographing information for allowing a document author to define the timing at which the first file support object and the second file support object are retrieved by a user, the choreographing information comprising data slices from the first file support object [interleaved with data slices from] and the second file support object multiplexed so as to incrementally [display] render the first file support object and the second file support object to the user at substantially the same time.

Claims 51 and 52 have been added.